Department copy

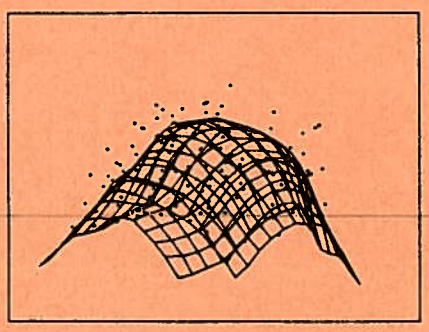
# SMART

## User's Guide

*Jerome H. Friedman*

**Technical Report No. 1**
**October 1984**

# Laboratory for Computational Statistics



**Department of Statistics**
**Stanford University**

Department copy

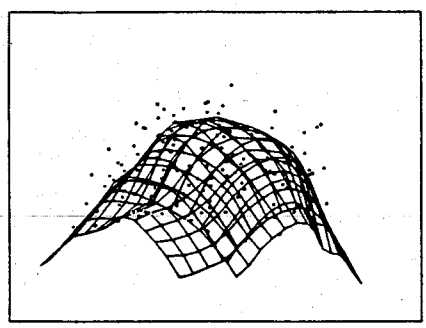
# SMART

**User's Guide**

*Jerome H. Friedman*

**Technical Report No. 1**
**October 1984**

## Laboratory for Computational Statistics



**Department of Statistics**
**Stanford University**

# SMART

## User's Guide

Jerome H. Friedman

Department of Statistics

and

Stanford Linear Accelerator Center

Stanford University

Abstract

This note describes software implementing the SMART algorithm. SMART generalizes the projection pursuit method to classification and multiple response regression. SMART also provides a more efficient algorithm for single response projection pursuit regression.

LCSØØ 1

# SMART

User's Guide

## 1. Introduction

SMART (Smooth Multiple Additive Regression Technique) is a method for modeling a set of response variables $Y_i$ $(1 \leq i \leq q)$ as functions of a set of predictor variables $X_j$ $(1 \leq j \leq p)$ based on matched observations (training data)

$$y_{1k}, y_{2k}, \cdots, y_{qk}, x_{1k}, x_{2k}, \cdots, x_{pk} \quad (1 \leq k \leq N) \tag{0}$$

the models take the form

$$E[Y_i \mid x_1, x_2, \cdots, x_p] = \overline{Y}_i + \sum_{m=1}^{M} \beta_{im} f_m(\sum_{j=1}^{p} \alpha_{jm} x_j) \tag{1}$$

with $\overline{Y}_i = EY_i$, $Ef_m = 0$, $Ef_m^2 = 1$ and $\sum_{j=1}^{p} \alpha_{jm}^2 = 1$. The coefficients $\beta_{im}, \alpha_{jm}$, and the functions $f_m$ are parameters of the model and are estimated by least squares. The criterion

$$L_2 = \sum_{i=1}^{q} W_i E[Y_i - \overline{Y}_i - \sum_{m=1}^{M} \beta_{im} f_m(\alpha_m^T x)]^2 \tag{2}$$

is minimized with respect to the parameters $\beta_{im}, \alpha_m^T = (\alpha_{1m} \cdots \alpha_{pm})$ and the functions $f_m$. The response weights $W_i$ $(1 \leq i \leq q)$, specified by the user, permit some flexibility in specification of the loss metric (see below). The expected values are computed from the data as

$$E[Z] = \sum_{k=1}^{N} w_k z_k / \sum_{k=1}^{N} w_k \tag{3}$$

where $Z$ is considered to be a random variable and $z_k$ $(1 \leq k \leq N)$ are its realized values in the data. The observation weights $w_k (1 \leq k \leq N)$, specified by the user, can be employed to assign differing mass to different observations. They can also be used to implement iterative reweighting schemes for robustification or approximate maximum likelihood fitting.

It should be noted that the loss criterion $L_2$ (2) is sensitive to the relative scales of the response variables $Y_i$ as is true for any distance measure. The influence of each $Y_i$ will

be in proportion to its variance $\mathrm{Var}(Y_i)$. If it is desired that each response variable have the same effect on the loss criterion one can set $W_i = 1/\mathrm{Var}\,(Y_i)$ or rescale the responses to have the same variance.

From (1) it is seen that SMART modeling is a generalization of projection pursuit regression PPR (Friedman and Stuetzle, 1981). Each response variable is modeled as a (usually) different linear combination of the predictor functions $f_m$. Each predictor function is taken as a (smooth, but otherwise unrestricted) function of a (usually) different linear combination of the predictor variables. Estimates for the parameters of the linear combinations, and the functions, are chosen to be the values that minimize the loss criterion $L_2$ (2). For the case of a single response variable ($q = 1$) SMART models have the same form as PPR models but they differ from PPR models in that SMART chooses estimates that minimize (2) whereas PPR choooses the $\alpha_m^T$ $(1 \leq m \leq M)$ in a forward stagewise manner. This can result in considerably different models, especially when there are high associations among the predictor variables.

## 2. Classification

Classification is closely related to regression. Here a single response variable $Y$ assumes several categorical (unorderable) values $(c_1, c_2, \cdots, c_q)$. The loss criterion is usually taken to be the misclassification risk

$$R = E[\min_{1 \leq j \leq q} \sum_{i=1}^{q} l_{ij}\, p(i \mid x_1, x_2, \cdots x_p)] \tag{4}$$

where $l_{ij}$ is the (user specified) loss for predicting $Y = c_j$ when its true value is $c_i$ $(l_{ii} \equiv 0)$. The conditional probability $p(i \mid x_1 \cdots x_p)$ is the probability that $Y = c_i$ given a particular set of values for the predictor variables $x_1 \cdots x_p$. The sum in (4) is simply the loss for predicting $Y = c_j$ given $x_1 \cdots x_p$. The minimization operation provides a decision rule that minimizes this loss at each set of predictor values. The risk is then the expected or average loss using this optimal decision rule. The art of classification is to find estimates of the conditional probabilities that minimize the misclassification risk.

Defining category (class) indicator variables for each observation $k$ as

$$h_{ik} = \begin{cases} 1 \text{ if } y_k = c_i & 1 \leq k \leq N \\ 0 \text{ otherwise} & 1 \leq i \leq q \end{cases}$$

one has

$$p(i \mid x_1 \cdots x_p) = \frac{\pi_i S}{s_i} E[H_i \mid x_1 \cdots x_p] \tag{5}$$

with $\pi_i$ the unconditional (prior) probability that $Y = c_i$ $(H_i = 1)$, $s_i = \sum_{k=1}^{N} w_k \delta(y_k, c_i)$, and
$S = \sum_{i=1}^{q} s_i$. Here $\delta$ is the Kronecker delta function

$$\delta(a, b) = \begin{cases} 1 \text{ if } a = b \\ 0 \text{ otherwise} \end{cases}$$

Substituting (5) into (4) one has

$$R = E\Big[\min_{1 \le j \le q} S \sum_{i=1}^{q} \frac{\pi_i l_{ij}}{s_i} E[H_i \mid x_1 \cdots x_p]\Big] \tag{6}$$

From this one sees that the optimal decision rule for a given set of predictor values $x_1 \cdots x_p$ is to assign $Y = c_{J^*}$ where $J^*$ is the integer value $(1 \le J^* \le q)$ that minimizes the sum in (6).

When the prior probabilities $\pi_i$ $(1 \le i \le q)$ are unknown, they can be estimated from the data as $\hat{\pi}_i = s_i/S$. Often the losses $l_{ij}$ are taken to be simply $l_{ij} = 1 - \delta(i, j)$. When both of these situations occur the misclassification risk reduces to simply the misclassification probability.

SMART models the condition expectations (5, 6) in form given by (1). Ideally the parameter and function estimates should be chosen to minimize the misclassification risk (6). However, as discussed in Breiman, Friedman, Olshen and Stone (1983) (see also Efron, 1978), this can lead to difficulties due to the non-convexity of (6). A good surrogate is the squared error loss criterion $L_2$ (2) with

$$W_i = \frac{S\pi_i}{s_i} \sum_{j=1}^{q} l_{ij}. \tag{7}$$

## 3. Modeling Strategy.

It is the purpose of the SMART algorithm to minimize $L_2$ (2) with respect to the parameters $\beta_{im}, \alpha_{jm}$, and functions $f_m$ $(1 \le i \le q,\ 1 \le j \le p,\ 1 \le m \le M)$, given the

training data (0) and a loss metric $W_i$ $(1 \leq i \leq q)$. The specifics of the algorithm are given in Appendix 1. The principal task of the user is to choose $M$ (2) the number of predictive terms comprising the model. Increasing the number of terms decreases the bias (model specification error) at the expense of increasing the variance of the (model and parameter) estimates. Since the expected squared error, ESE, is the sum of these two effects - ESE = (bias)$^2$ + variance, there is an optimal value for $M$. Sample reuse techniques can be used to estimate these effects - ESE through cross-validation (Stone, 1977) and (Geisser, 1975), and variance through bootstrapping (Efron, 1983). It is possible to implement these procedures in conjunction with SMART with the aim of estimating an optimal value for $M$ as well as confidence intervals for estimates.

Since the variance tends to increase linearly with increasing $M$ while the (bias)$^2$ tends to drop rapidly for small (increasing) $M$, leveling off to a slow decrease for larger $M$, a good estimate for the optimal $M$ value can usually be made by simply inspecting $L_2$ vs. $M$ for various values of $M$. That point at which a unit decrease in $M$ leads to a relatively large increase in $L_2$ (compared to that for close-by larger $M$ values) is often a good choice. Since the ESE tends to vary slowly as a function of $M$ in the region near the optimal $M$ value (especially on the side of increasing $M$), the choice is not critical provided it is not too small.

For a given value of $M$, solutions (minimizing $L_2$) may not be unique. Sometimes there are local minima that can trap the SMART algorithm thereby masking a better global minimum. Such local minima represent solutions that are relevant to larger (higher $M$) models. Solutions are not necessarily found in optimal order as $M$ is increased. This suggests a backwards stepwise model selection procedure.

The strategy is to start with a relatively large value of $M$ (say $M = M_L$) and find all models of size $M_L$ and less. That is, solutions that minimize $L_2$ are found for $M = M_L,\ M_L - 1,\ M_L - 2, \cdots,\ 1$ in order of decreasing $M$. The starting parameter values for the numerical search in each $M$-term model are the solution values for the $M$ most important (out of $M + 1$) terms of the previous model. Term importance is measured as

$$I_m = \sum_{i=1}^{q} W_i \mid \beta_{im} \mid \quad (1 \leq m \leq M) \tag{8}$$

normalized so that the most important term has unit importance.
(Note that the variance of all $f_m$ is one.) The starting point for the minimization of the largest model, $M = M_L$, is given by an $M_L$ term stagewise model (Friedman and Stuetzle, 1981).

The sequence of solutions generated in this manner is then examined by the user and a final model is chosen according to the guidelines above.

## 4. Relative Importance of Predictor Variables

It is often useful to have an idea of the relative importance of each predictor variable to the final model. For (single response) linear models an often used measure is the absolute value of the corresponding regression coefficient $\alpha_j$ times a scale measure of the predictor variable $\sigma_j$, $I_j = \sigma_j \mid \alpha_j \mid$, $(1 \leq j \leq p)$. A corresponding relative importance measure for (multiple response) nonlinear models would be

$$I_j = \sigma_j \sum_{i=1}^{q} W_i \, E \mid \frac{\partial \hat{Y}_i}{\partial X_j} \mid \quad (1 \leq j \leq p)$$

with $\hat{Y}_i = E[Y_i \mid x_1 \cdots x_p]$. For SMART models (1) this becomes

$$I_j = \sigma_j \sum_{i=1}^{q} W_i \, E \mid \sum_{m=1}^{M} \beta_{im} \alpha_{jm} f'(\alpha_m^T x) \mid \quad (1 \leq j \leq p) \tag{9}$$

where $f_m'(z) = df_m/dz$. In the case of only one term, $M = 1$, (9) is equivalent to $I_j = \sigma_j \mid \alpha_j \mid$. It is important to keep in mind that the same care is required in interpreting (9) as in the corresponding interpretation of regression coefficients in linear models, especially in the presence of high collinearity among the predictor variables.

## 5. SMART Software - Input

SMART software is implemented as a collection of FORTRAN subroutines. The user interface is provided by the parameter list (calling sequence) to some of these routines. In order to apply SMART modeling it is necessary to write a driver program that reads the training data (0) into internal storage arrays, sets various parameters of the problem, and then pass these to SMART through the parameter list of the appropriate SMART subroutine.

5.1 SMART Regression

CALL SMARTR (ML, MU, P, Q, N, W, X, Y, WW,
SMOD, NSMOD, SP, NSP, DP, NDP)

The first nine parameters are input defining the problem and the last six define storage workspace necessary for the operation of the program.

The first two parameters $ML, MU$ (type integer) define the sequence of models in the backwards stepwise model selection procedure described above (Section 4). The value of the first parameter, $ML$, defines the number of terms ($M$ in (1)) in the largest model of the sequence while the second similarily defines the smallest model in the sequence. This smallest $MU$-term model is the one stored (in SMOD, see below) for later predictive use. (Also the predictive functions $f_m(\alpha_m^T x)$ are only returned for the $MU$-term model and relative predictor variable importances (9) are calculated only for this model.) A good strategy is to initially set $ML$ reasonably large (subject to computing time limitations) and set $MU = 1$, thereby generating all models of size $M = ML$ and less (1). The particular model selected by the user can then be computed and stored for later predictive use. Also, the predictive functions and relative predictor variable importances (9) can be inspected for this model. This is accomplished by rerunning the program with the same value for $ML$ but with $MU$ set to the size of the user selected model.

The next three parameters (type integer) define the size of the problem:

$P =$ number of predictor variables

$Q =$ number of response variables

$N =$ number of observations.

these correspond to the quantities $p, q, N$ of (2).

The next four parameters (type real) contain the data and corresponding weights. These are arrays that must be declared and appropriately dimensioned in the user written driver program:

Real $W(N)$: observation weights

$W(K)$ contains the weight for the $Kth$ observation($w_k$ in (3))

Real $X(P,N)$: predictor data matrix

$X(J,K)$ contains the value of the $Jth$ predictor variable of $Kth$ observation.

Real $Y(Q,N)$: response data matrix

$Y(I,K)$ contains the value of the $Ith$ response variable of $Kth$ observation.

Real $WW(Q)$: response weights

$WW(I)$ contains the response weight for $Ith$ response ($W_i$ in (2)).

The final six parameters define storage workspace necessary for the operation of the algorithm. The parameters SMOD, SP, and DP are arrays that must be declared and dimensioned in the calling program. The quantities NSMOD, NSP, and NDP (type integer) give the dimensions assigned (by the user) to the corresponding three storage arrays so that SMART can check if the workspace sizes (dimensioned values) are large enough.

REAL SMOD (NSMOD): stores parameters of the final ($M = MU$) term model. Minimum dimension is $\underline{NSMOD \geq ML(P+Q+2N)+Q+7}$.

REAL SP(NSP): single precision scratch storage workspace. Minimum dimension is $\underline{NSP \geq N(Q+15)+Q+3P}$.

DOUBLE PRECISION DP(NDP): double precision scratch storage workspace. Minimum dimension is $\underline{NDP \geq P(P+1)/2+6P.}$

### 5.2 SMART Classification

```
CALL SMARTC  (ML, MU, P, Q, N, W, X, C, PI, FLS,
              SMOD, NSMOD, SP, NSP, DP, NDP)
```

The first ten parameters are input defining the problem and the last six define storage workspace. The parameters $ML, MU, P, N, W, X, DP, NDP$ are identical to the corresponding ones for regression and are described in Section 5.1.

The parameter $Q$ (type integer) gives the number of classes - the number of distinct values for the categorical response variable, $C$. The array $C$ (type real) gives the class identity of each observation:

REAL $C(N)$:

$C(K)$ contains the value of class
label for $Kth$ observation
$(1.0 \leq C(K) \leq \text{FLOAT}(Q)),\ K = 1, N.$

The next two parameters (type real) define the prior (uncondition) probabilities and the loss structure for the classification problem.

Real $PI(Q)$: class priors

$PI(I)$ contains the prior probability for $Ith$
class ($\pi_i$) in (5) and (7)
$(PI(I) > 0,\ I = 1, Q,$ and $\sum_{I=1}^{Q} PI(I) = 1).$

Real $FLS(Q, Q)$: loss matrix

$FLS(I, J)$ contains the loss for
misclassifying a class $I$ observation
as class $J$ ($l_{ij}$ in (4) and (6)).
$(FLS(I, J) \geq 0,\ I \neq J,$ and
$FLS(I, I) = 0,\ I = 1, Q,\ J = 1, Q.)$

Often the prior probabilities $\pi_i$ $(1 \leq i \leq q)$ are unknown and are to be estimated from the data as $\hat{\pi}_i = s_i/S$ where $s_i$ is the sum of (observation) weights for the class $i$ observations and $S$ is the sum of weights for all observations. When this is the case, there are no user defined prior probabilities and the $PI$ array need not be declared or dimensioned in the calling program. This situation is indicated by passing a single scalar value $BIG$ in place of the $PI$ array in the corresponding position in the parameter list. The value of $BIG$ is a large number defined internally to SMART - see section 5.3

Similarly, only a simple loss structure is often desired, namely $l_{ij} = 1 - \delta(i, j)$. That is, a simple unit loss for each misclassified observation. When this is the case the $FLS$ array need not be declared or dimensioned in the calling program, and the single value

*BIG* entered its place in the parameter sequence.

The storage workspace arrays SMOD and SP, and their corresponding array dimensions NSMOD and NSP, have the same meaning as for the regression problem (Section 5.1), however, the size of the dimensions must be a little larger for classification:

REAL SMOD(NSMOD), SP(NSP)

NSMOD $\geq \underline{ML(P+Q+2N)+2Q+7}$

NSP $\geq \underline{N(Q+15)+2Q+3P}$.

### 5.3 Incidental Parameters

COMMON/PARMS/IFL, LF, SPAN, ALPHA, BIG

This labeled common contains internal parameters of the algorithm that the user may wish to change. Default values for these parameters are set at compile time in a BLOCK DATA subprogram. This labeled common need only be declared in the user's calling program if he wishes to change any of their values from the default settings. This can be done using executable assignment statements in the user routine in which this common is declared.

| | |
|---|---|
| INTEGER IFL: | FORTRAN file number for writing printed output (Default, IFL=6) If $IFL \leq 0$ no printed output will be generated. |
| INTEGER LF: | Optimizing level for minimization algorithm, $0 \leq LF \leq 3$ (default, LF=2). This controls tradeoff between speed and thoroughness of optimization algorithm (See Appendix 1). |
| REAL SPAN, ALPHA: | Super smoother parameters. These control operation of smoother used to obtain function estimates (See Friedman, 1984) (Default values, SPAN=0.0, ALPHA =0.0.) |
| REAL BIG: | A large representative floating point number very much larger than the largest possible (absolute) data value (Default, $BIG = 10^{20}$) |

## 6. SMART Software - Output

The primary output of SMART is a model for estimating $E[Y_i \mid x_1 \cdots x_p]$ and, for classification, a decision rule (6). A second level of output would be the parameters of the SMART model (1). These are useful for interpretation of the dependence of each $Y_i$ on $(x_1 \cdots x_p)$. A third level of output would be additional diagnostic information (8), (9) useful for interpretation.

### 6.1 Primary Output

For a given set of values comprising a predictor vector $(x_1 \cdots x_p)$, the SMART estimate (1) for $\hat{Y}_i(x_1 \cdots x_p) = E[Y_i \mid x_1 \cdots x_p]$ is obtained by executing

CALL YHAT (XT, SMOD, YH).

This statement must be executed after calling either SMARTR (Section 5.1) or SMARTC (Section 5.2). The array SMOD must be the same as in the call to SMARTR or SMARTC. The quantities $XT$ and $YH$ have the following meaning:

Real XT(P): input predictor vector

XT(J) contains the value of $Jth$ variable, $x_j$.

Real YH(Q): output expected response values (given $XT$)

YH(I) contains the estimated expected value for Ith response,

$\hat{Y}_i(x_1 \cdots x_p)$.

For classification the minimum (estimated) risk decision rule (6) is obtained by executing

CALL CLSFY (PI, FLS, YH, SMOD, ICL, RSK).

This statement must be executed after calling both SMARTC and YHAT. The quantities PI, FLS SMOD are described in Section 5.2 and must be the same as in the call to SMARTC. The array $YH$ is the output response vector from YHAT (see above, this section) giving the relative class probabilities given $XT = (X_1, X_2 \cdots X_P)$. The two output quantities (from CLSFY) are ICL and RSK. The first, ICL, contains as its value the class assignment that minimizes the (estimated) misclassification risk, while RSK has the estimated value of this minimum risk. This second quantity is useful in accessing the relative confidence in the class assignment.

SMART modeling (as implemented here) does not constrain the values of the response conditional expectation estimates. When these expectations are interpreted as conditional probabilities (as in the case of classification), it is useful to constrain their individual values so that the corresponding conditional probabilities (5) are between zero and one and sum to one. When these constraints are violated by the $\hat{Y}_i$ as output from YHAT, CLSFY modifies their values to satisfy these constraints in a way that preserves the relative values of the corresponding probabilities $p_i$ (5). The conditional probability values are stored in YH (replacing the conditional expectation estimates) before returning to the calling program.

### 6.2 Secondary Output

The parameters of the SMART model are packed in the SMOD array upon return from SMARTR or SMARTC. Several user callable subroutines are available to obtain them in a convenient form under program control. In addition some of these parameters, $M$, $\alpha_{jm}$, $\beta_{im}$ $(i = 1, q,\ j = 1, p,\ m = 1, M)$ (1), appear on the standard (printer) output file IFL (provided IFL $> 0$, see section 5.3). The functions $f_m$ $(m = 1, M)$ (1) do not appear on the standard output file. They must be obtained under program control, as described below, and then transferred to a local (installation dependent) graphics library for representation on a graphical output device. These functions are available only for the final MU-term model (see Section 5.1).

The value of the goodness-of-fit criterion, $FL2$, $L_2$ (2) for the final MU-term model can be obtained by executing the statement

FL2 = GOF(SMOD, MU).

The array SMOD must be the same as in the call to SMARTR (Section 5.1) or SMARTC (Section 5.2). The output quantity, $MU$, is the number of terms of the final (user specified) model.

The parameter vectors $\alpha_{jm}$ $(1 \leq j \leq p)$ and $\beta_{im}$ $(1 \leq i \leq q)$ can be obtained for each term, $m$, by executing the statement

FP = GTPRMS(ITERM, SMOD, A, B).

The array SMOD must be the same as in the call to SMARTR or SMARTC. The input

quantity ITERM is the term number for which the parameters are desired. The parameters are stored in the real arrays A and B:

REAL A(P): parameters of predictor linear combinations
A(J)= $\alpha_{jm}$ $(J, j = 1, p)$

REAL B(Q): parameters of the response linear combinations
B(I) =$\beta_{im}$ $(I, i = 1, q)$

and ITERM = m.

The quantity FP has the value 1.0 if $1 \leq$ ITERM $\leq$ MU and 0.0 otherwise. If FP = 0.0, then no values are stored in the output arrays.

Each function $f_m(\alpha_m^T x)$ is represented by a set of matched pairs $(f_{mk}, t_{mk}), 1 \leq k \leq N$, one pair for each observation. Here $f_{mk} = f_m(t_{mk})$ with $t_{mk} = \alpha_m^T x_k$. These pairs can be plotted as points on an available graphics device with $f_m$ as the ordinate and $t_m$ the abscissa. The points representing the function for each term, $m$, are obtained by executing the statement

FP = GTFUN(ITERM, SMOD, F, T).

The quantities FP, ITERM, SMOD have the same meaning as with GTPRMS described in the preceding paragraph. The function is stored in the two output arrays:

REAL F(N): ordinates, F(K) is the ordinate value for the $Kth$ observation

Real T(N): abscissas, T(K) is the abscissa value for the $Kth$ observation

where K=1, N. (Note that the observations are labeled here in increasing order of $T(K)$ rather than in their order in the data matrix.)

### 6.3 Third Level Output

In addition to the output obtainable under program control, SMART also writes information to the standard output file IFL (provided IFL>0, see Section 5.3). This information can help with model selection and in interpretation of the selected final model. For this output the goodness-of-fit is always expressed in terms of fraction of unexplained variance defined as

$$e^2 = L_2 / \sum_{i=1}^{q} W_i E[Y_i - \overline{Y}_i]^2 \qquad (10)$$

with $\overline{Y}_i = EY_i$ and $L_2$ given by (2). Note, however, that FUNCTION GOF (Section 6.2) returns the value of $L_2$, not $e^2$, for the final model.

This third level output consists of a listing of all of the ($M$–term) solutions $MU \leq M \leq ML$, in order of decreasing value of $M$. Each solution is represented by the $\{\alpha_{jm}, \beta_{im}\}$, $(1 \leq i \leq q,\ 1 \leq j \leq p)$, for each term $m$ $(1 \leq m \leq M)$. The $\alpha_{jm}$ are given in order of increasing $j$ $(A =)$ and the $\beta_{im}$ in order of increasing $i(B =)$. The value of $e^2$ for a solution precedes the parameter listings of its terms.

Following the term parameter listings of a particular solution is a listing of the relative importance of each term $I_m$ (8) $(1 \leq m \leq M)$. The starting parameter values in the numerical search for the next smaller $(M-1$ term) model are the solution values for the $M-1$ most important terms of this ($M$-term) model.

Following the relative term importance listing is a listing of the fraction of unexplained variance $e_i^2$ for each response $Y_i$ $(1 \leq i \leq q)$ separately. Here

$$e_i^2 = E[Y_i - \overline{Y}_i - \sum_{m=1}^{M} \beta_{im} f_m(\alpha_m^T x)]^2 / E[Y_i - \overline{Y}_i]^2.$$

this output does not appear if there is only one response variable $(q = 1)$.

For classification there are two additional quantities listed with each ($M$–term) solution. These are two different estimates of the misclassification risk associated with using this model for the conditional expectations in a minimum risk decision rule (6). The first estimate $R_1$ (MISCLASSIFICATION RISK) is obtained by classifying each training observation $k$ $(1 \leq k \leq N)$ using the minimum loss rule (6)

$$J_k^* = \min_{1 \leq j \leq q}{}^{-1} \{\sum_{i=1}^{q} \frac{\pi_i l_{ij}}{s_i} E[H_i \mid x_{1k} \cdots x_{p_k}]\} \tag{11}$$

and then computing the risk by averaging the loss associated with the resulting misclassifications

$$R_1 = \sum_{k=1}^{N} w_k \sum_{i=1}^{q} \frac{\pi_i}{s_i} l_{iJ_k^*}. \tag{12}$$

The second estimate $R_2$ (CALCULATED FROM PROBABILITY ESTIMATES) is the value of $R$ (6) computed by substituting the conditional expectation estimates of this ($M$–term) model directly into (6).

To the extent that the conditional expectation (probability) estimates are accurate these two risk estimates should have similar values. Note that $R_1$ is nearly always less than $R_2$. However, it is often possible to do accurate classification in the presence of very poor probability estimates. This is especially true for the simple loss structure $l_{ij} = 1 - \delta(i,j)$ where it is only necessary to correctly estimate which class has the highest probability given $x_1 \cdots x_p$. The probability values themselves or even their order (except for the largest) are not needed in this case.

Comparing the values of $R_1$ and $R_2$ gives some indication of how well the model conditional expectation estimates are approximating the true underlying probabilities. If $R_1$ is much smaller than $R_2$ (which is often the case) then the probability estimates are not too close. In this case some care should be exercised in interpreting the values of $YH(I)$ $(1 \leq I \leq Q)$ and RSK as returned by CLSFY (see Section 6.1).

The quantities described (so far) in this section are listed for each $M$– term solution $(MU \leq M \leq ML)$. The final $MU$–term solution is the SMART model relevant to the output described in Sections 6.1 and 6.2. The relative importance of each predictor variable $I_j$ $(1 \leq j \leq p)$ (9) (standardized so that the most important variable has unit importance) is also computed for this last $MU$–term SMART model and listed at the end of the standard output.

## Appendix I

### Numerical optimization of least squares criterion for SMART models

This section discusses the minimization of $L_2$ (2) simultaneously with respect to $\alpha_{jm}$ $(1 \leq j \leq p), \beta_{im}$ $(1 \leq i \leq q)$ and the functions $f_m (1 \leq m \leq M)$ for a given number of terms $M$. An alternating optimization strategy is used. The parameters are grouped such that the solution for those in each group is straightforward given fixed values for those outside the group. A solution is obtained for the variables in the group and these solution values replace their current parameter values. Attention is then focused on the next group and this process repeated for its parameters. After solutions have been obtained for all groups of parameters, another pass is made over the groups obtaining new solution values given the new values for the parameters outside each group obtained in the previous pass. These passes are repeated until the loss criterion $L_2$ (2) fails to decrease on two consecutive passes. Usually a threshold $\epsilon$ is set at a small value and if improvement on two consecutive passes is less than $\epsilon$, iterations are stopped and the parameter values at that point taken as the solution. Since at each step in this process $L_2$ is made smaller through a partial minimization, and $L_2 \geq 0$, the alternating optimization must converge (provided $\epsilon$ is large compared to the numerical accuracy of the computer's arithmetic). However, there is no guarantee that the solution is the global minimum of $L_2$. It may be a local minimum. Strategy for dealing with this problem in the context of SMART modeling is discussed in Section 3.

The parameter grouping used in the SMART algorithm is hierarchical. The first level grouping is by term. The parameters $\alpha_{jm} (1 \leq j \leq p)$, $\beta_{im} (1 \leq i \leq q)$ and the function $f_m$ (for fixed $m$) form each group. There are obviously $M$ such groups. At the second level the parameters of each term are divided into three groups: the $\alpha_{jm}$ $(1 \leq j \leq p)$ form the first (sub) grouping, the $\beta_{im}$ $(1 \leq i \leq q)$ form the second and the function $f_m$ forms the third.

Consider a particular term, $k$ $(1 \leq k \leq M)$. The loss criterion (2) can be reexpressed

as

$$L_2 \equiv L_2^{(k)} = \sum_{i=1}^{q} W_i \; E[R_{i(k)} - \beta_{ik} f_k(\alpha_k^T x)]^2 \qquad (A1)$$

with

$$R_{i(k)} = Y_i - \overline{Y}_i - \sum_{m \neq k} \beta_{im} f_m(\alpha_m^T x) \qquad (A2)$$

Equation A1 isolates the *kth* term's contribution to the criterion. Following the alternating optimization strategy we minimize $L_2$ ($L_2^{(k)}$) with respect to the parameters of the *kth* term. These parameter values are then used to help define $R_{i(k')}, k' \neq k$, to obtain new solutions for the parameters of other terms. Repeated passes are made over all the terms until convergence ($L_2$ stops decreasing–see above).

We now focus on obtaining solutions for the parameters of the *kth* term given $R_{i(k)}$ (A2). The solutions for the $\beta_{ik}$ (given $f_k$ and $\alpha_k^T$) are straightforward

$$\beta_{ik}^* = \frac{E\left[R_{i(k)} f_k(\alpha_k^T x)\right]}{E\left[f_k(\alpha_k^T x)\right]^2} \quad (1 \leq i \leq q) \qquad (A3)$$

(Remember that $E[R_{i(k)}] = E[f_k(\alpha_k^T x)] = 0$.)

The solution for the function $f_k$ (given $\beta_{ik}$ and $\alpha_k^T$) is almost as easily obtained. Reexpressing $L_2^{(k)}$ (A1) as

$$L_2^{(k)} = E_{\alpha_k^T x} \, E[\sum_{i=1}^{q} W_i (R_{i(k)} - \beta_{ik} f_k)^2 \mid \alpha_k^T x], \qquad (A4)$$

we see that it is minimized if $f_k$ is chosen to minimize the conditional expectation in A4 for each value of $\alpha_k^T x$. This is accomplished by

$$f_k^*(\alpha_k^T x) = E[\sum_{i=1}^{q} W_i \beta_{ik} R_{i(k)} \mid \alpha_k^T x] / \sum_{i=1}^{q} W_i \beta_{ik}^2 \qquad (A5)$$

Since we require $Ef_k = 0$ and $Ef_k^2 = 1$, we standardize $f_k^*$, making the denominator in (A5) irrelevant.

It remains to find a solution that minimizes $L_2^{(k)}$ (A1) with respect to $\alpha_k^T = (\alpha_{1k}, \alpha_{2k}, \cdots \alpha_{pk})$ given values for $\beta_{ik}$ $(1 \leq i \leq q)$ and a (fixed) function $f_k$. Unlike the other parameters ($\beta_{ik}$ and $f_k$), $\alpha_k^T$ does not enter in a purely quadratically way into the loss

criterion. Therefore, solutions may not be unique, and they cannot be obtained in a single step. An iterative numerical optimization must be performed.

The loss criterion $L_2$ (2, A1) can be expressed in the generic form

$$L_2(\alpha_k) = \sum_{i=1}^{q} W_i E[g_i(\alpha_k)]^2 \tag{A6}$$

with

$$g_i(\alpha_k) = \left(R_{i(k)} - \beta_{ik} f_k(\alpha_k^T x)\right) \tag{A7}$$

The classical numerical optimization technique for criteria of the form (A6) is the Gauss-Newton method (see Gill, Murray and Wright, 1981, Section 4.7). Let $\alpha_k^{(0)T} = (\alpha_{1k}^{(0)}, \cdots, \alpha_{pk}^{(0)})$ be a trial set of values at some point during the optimization. The Gauss-Newton estimate for the solution $\alpha_k^T$ (the next set of trial values in the iterative process) is $\alpha_k^T = \alpha_k^{(0)T} + \Delta^T$ where the vector $\Delta^T$ is the solution to the set of simultaneous equations

$$\sum_{i=1}^{q} W_i E[(\frac{\partial g_i}{\partial \alpha_k})^T (\frac{\partial g_i}{\partial \alpha_k})]\Delta = -\sum_{i=1}^{q} W_i E[(\frac{\partial g_i}{\partial \alpha_k})^T g_i] \tag{A8}$$

The function $g_i$ and the vector of partial derivatives are evaluated at $\alpha_k^{(0)}$. From A7 one has

$$\frac{\partial g_i}{\partial \alpha_k}(\alpha_k^{(0)}) = -\beta_{ik} f_k'(\alpha_k^{(0)T} x)x \tag{A9}$$

where $f'(z) = df/dz$. After solving (A8) for $\Delta$, $\alpha_k$ replaces $\alpha_k^{(0)}$ and the process can be repeated until convergence ($L_2$ stops decreasing).

It is possible that a Gauss-Newton step fails to decrease $L_2$ ($L_2(\alpha_k^{(0)} + \Delta) \geq L_2(\alpha_k^{(0)})$). In this case the step is cut in half ($\alpha_k = \alpha_k^{(0)} + \Delta/2$). If this new step still results in an increase in $L_2$, the step is cut again ($\alpha_k = \alpha_k^{(0)} + \Delta/4$). This repeated cutting of the step is continued until $L_2$ decreases. Since the matrix on the left-hand-side of (A8) is positive definite, $\hat{\Delta} = \Delta / \mid \Delta \mid$ is a valid descent direction and at some point the step halving must give rise to a decrease in $L_2$ (unless $\alpha_k^{(0)}$ represents a minimum of $L_2$).

As discussed in Section 6.2, the functions $f_k(\alpha_k^T x)$ are stored as an ordinate and abscissa value for each observation. The derivative estimates $f_k'(\alpha_k^T x)$ are similarily stored (see below). These values are obtained when $f_k(\alpha_k^T x)$ is evaluated (A5). When $\alpha_k^{(0)T}$ is

changed to $\alpha_k^T$ (via Gauss-Newton update), an interpolation scheme must be employed to obtain values for $f_k$ $(\alpha_k^T x)$ from $f_k(\alpha_k^{(0)T} x)$. This interpolation is almost as expensive as obtaining the optimal function for the new argument $\alpha_k^T x$. We, therefore, do not iterate the Gauss-Newton stepping until convergence for a given function, but rather take only a single step. A new (optimal) function $f_k^*[(\alpha_k^{(0)T} + \Delta^T)x]$ (A5) is evaluated, and the next Gauss-Newton step (A7-A9) is made based on this new function. Step cutting, as described above, is employed for bad steps. In this way both the function and the predictor linear combination for the $k-th$ term are simultaneously optimized by the Gauss-Newton iteration procedure.

The expected values $E[\cdot]$ are easily evaluated via (3). The conditional expectation estimates (A5) for evaluation of the optimal functions are more difficult. The method used here is described in detail in Friedman (1984). The derivative estimates (9 and A9) are made by taking first differences of the function estimates

$$f_k'(\alpha_k^T x_l) = \frac{[f_k(\alpha_k^T x_{l+1}) - f_k(\alpha_k^T x_{l-1})]}{\alpha_k^T (x_{l+1} - x_{l-1})} \quad (2 \leq l \leq N-1) \qquad (A10)$$

where the $x_l$ are labeled in increasing order of $\alpha_k^T x$. Endpoints ($l = 1$ and $l = N$) are handled by simply copying the values of their nearest neighbors. Such estimates can become unstable if the denominator becomes too small. This can be avoided by pooling observations for which

$$| \alpha_k^T (x_l - x_{l'}) | \leq \epsilon I \quad (1 \leq l, l' \leq N) \qquad (A11)$$

into a single observation for the purpose of derivative calculation. Here $I$ is the semi-interquartile range of $\alpha_k^T x$ and $\epsilon$ is a small number ($\epsilon \simeq 0.05$). This pooling can be done rapidly by using a method similar to the pooled-adjacent-violators algorithm for isotone regression (Kruskal, 1964).

The SMART program provides the user some control over the optimization process. This control is exercised through the parameter $LF$ in the $/PARMS/$ common block (see Section 5.3). This parameter can take integer values between zero and three (default, $LF = 2$). Level three ($LF = 3$) optimization is that which is described above in this section. The other (lower) levels induce some shortcuts in the optimization procedure.

Level two ($LF = 2$) optimization has the same goal as level three: that is, minimize $L_2$ (2) with respect to all parameters and functions. The two levels differ only in strategy. With level two only one Gauss-Newton step (for the $\alpha_{jk}$, $1 \leq j \leq p$) and corresponding function $f_k$ optimization is performed for each new set of $\beta_{ik}$ ($1 \leq i \leq q$) in the iteration loop for the $kth$ term (see A1), rather than completely optimizing with respect to $\alpha_{jk}$ and $f_k$ for each new set of $\beta_{ik}$. (For single response regression ($q = 1$) the two strategies are equivalent.) Level two optimization is usually faster but a little less robust (more easily trapped at saddle points) than level three.

The two lowest levels of optimization, levels zero and one, actually construct different models. These models have the same form as SMART models (1) but the parameter and function estimates are obtained by partially rather than completely minimizing $L_2$ (2). Level zero ($LF = 0$) implements a purely stagewise optimization strategy. At each stage the loss criterion $L_2$ (2) is minimized only with respect to the parameters and function of the $Mth$ term, $\beta_{iM}$ ($1 \leq i \leq q$), $\alpha_{jM}$ ($1 \leq j \leq p$), $f_M(\alpha_M^T x)$, given the previously established values for the corresponding quantities in the earlier terms ($1 \leq m \leq M-1$). The $M$ term model consists of the newly established estimates at the $Mth$ stage as well as those for the previous ($M-1$ term) model.

Level one optimization ($LF = 1$) represents a compromise between a purely stagewise strategy (level zero) and complete least squares (levels two and three). Here the estimates for the predictor linear combinations ($\alpha_m^T$) are obtained in a stagewise manner as described above. However, the $M$ term model is obtained by completely minimizing $L_2$ (2) with respect to the $\beta_{im}$ ($1 \leq i \leq q$) and $f_m$, given the stagewise estimates for the $\alpha_m^T$ ($1 \leq m \leq M$). For a single response ($q = 1$), level one optimization is similar to the procedure employed in PPR modeling (Friedman and Stuetzle, 1981). The only differences are in the backwards stepwise model selection procedure (see Section 3) used by SMART, and in the use of the Gauss-Newton (rather than a Rosenbrock) procedure for the numerical optimization.

The principal advantage of level zero or one optimization over complete least squares (levels two or three) is computation speed. If this is a problem, then the lower optimization levels can be used to rapidly obtain models that are often quite competitive with the

full least squares solution. This is especially true when there is not a high degree of association among the predictor variables. Also, the lower optimization levels can be useful for exploratory work.

## Acknowledgment

The use of the Gauss-Newton method in the context of projection pursuit optimization was suggested by Andreas Buja and an early implementation was made by Ronald Thristed.

## References


Breiman, L., Friedman, J.H., Olshen R.A., and Stone, C. J. (1983) Classification and Regression Trees. Wadsworth International, Belmont, California.

Efron, B. (1978) Regression and ANOVA with zero-one data: measures of residual variation. J. Amer. Statist. Assoc., 73, p. 113-121.

Efron, B. (1983) Estimating the error rate of a prediction rule: improvements on cross validation. J. Amer. Statist. Assoc., 78, p. 316-331.

Friedman, J.H. and Stuetzle, W. (1981) Projection pursuit regression. J. Amer. Statist. Assoc., 76, p. 817-823.

Friedman, J. H. (1984) A variable span scatterplot smoother. Stanford University, Department of Statistics, Report LCS 05.

Geisser, S. (1975) The predictive sample reuse method with applications. J. Amer. Statist. Assoc., 70, p. 320-328.

Gill, P.E., Murray, W. and Wright, M.H. (1981) Practical Optimization. Academic Press, San Francisco.

Kruskal, J.B. (1964). Nonmetric multidimensional scaling: a numerical method. Psychometrika 29, p. 115-129.

Stone, M. (1977). Cross-validation: a review. Math Operationforsch. Statist. Ser. Statist., 9, p. 127-139.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | |
|---|---|
| 1. REPORT NUMBER: LCS 1 | 2. GOVT ACCESSION NO. |
| 3. RECIPIENT'S CATALOG NUMBER | |
| 4. TITLE *(and Subtitle)*: SMART User's Guide | 5. TYPE OF REPORT & PERIOD COVERED: Technical |
| 6. PERFORMING ORG. REPORT NUMBER | |
| 7. AUTHOR(s): Jerome H. Friedman | 8. CONTRACT OR GRANT NUMBER(s): N00014-83-K-0472 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS: Department of Statistics, Stanford Linear Accelerator Center, Stanford University, Stanford, CA 94305 | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 11. CONTROLLING OFFICE NAME AND ADDRESS | 12. REPORT DATE: October 1984 |
| | 13. NUMBER OF PAGES: 24 |
| 14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* | 15. SECURITY CLASS. *(of this report)* |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author and should not be construed as an official Department of the Navy position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

projection pursuit method, multiple response regression, non parametric regression, classification, discriminant analysis

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This note describes software implementing the SMART algorithm. SMART generalizes the projection pursuit method to classification and multiple response regression. SMART also provides a more efficient algorithm for single response projection pursuit regression.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*